

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:

Date: April 16, 2007

Robbert C. VAN DER LINDEN, et al.

Confirmation No. 4213

Serial No: 10/648,499

Group Art Unit: 2165

Filed: August 25, 2003

Examiner: Mark A. RADTKE

For: METHOD AND SYSTEM FOR UTILIZING A CACHE FOR PATH-LEVEL
ACCESS CONTROL TO STRUCTURED DOCUMENTS STORED IN A
DATABASE

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sir or Madam:

Appellant submits this Appeal Brief pursuant to the Notice of Appeal filed in this case on December 19, 2006.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corp. of Armonk, New York by virtue of an assignment from the inventor(s) recorded in the U.S. Patent and Trademark Office on August 25, 2003, at Reel No. 014443, Frame No. 0362.

II. RELATED APPEALS AND INTERFERENCES

There are no appeals, interferences, or judicial proceedings known to Appellant, the Appellant's legal representative, or Assignee, which may be related to, directly affect, be directly

affected by, or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-31 have been rejected. Appeal is taken from the rejection of claims 1-31.

IV. STATUS OF AMENDMENTS

No amendments were filed subsequent to the final Office action dated September 19, 2006.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present invention is directed to “method and system for providing path-level access control evaluation for a structured document” (pg. 20, lns. 11-12). Access control evaluation is improved by having a database management system (DBMS) evaluate “one value expression . . . for a path . . . instead of a plurality of access control rules” (pg. 20, lns. 14-15), which make evaluation more efficient. “In addition, by temporarily storing the evaluation result of a data-independent value expression in [a] cache . . ., the number of evaluations the DBMS . . . performs is minimized, further optimizing the access control evaluation process. Moreover, . . . access control evaluation is performed during run time, as opposed to compile time, thereby allowing an administrator to change the access control policy at run time without having to recompile the query” (pg. 20, lns. 16-21).

Independent claim 1 recites a method for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and

each of the plurality of nodes is described by a path. The method includes storing an access control statement in a cache entry (806a, 807a, 808a, 809a) for a path (806, 807, 808, 809) associated with a node of the plurality of nodes. *See, e.g.*, pg. 5, ln. 20 to pg. 6, ln. 2; pg. 18, lns. 14-18; figs. 1 and 8. The method also includes receiving a query, wherein the query comprises a request to access the node (702). *See, e.g.*, pg. 18, lns. 4-10; fig. 7A. In addition, the method includes checking the cache entry (806a, 807a, 808a, 809a) for the path (806, 807, 808, 809) associated with the node (704). *See, e.g.*, pg. 18, lns. 11-19; figs. 7A and 8. The method further includes granting or denying access to the node based on the access control statement in the cache entry (806a, 807a, 808a, 809a) for the path (806, 807, 808, 809) associated with the node (708, 718, 719, 730, 732). *See, e.g.*, pg. 18, ln. 20 to pg. 19, ln. 3; pg. 19, lns. 6-8; figs. 7A, 7B, and 8.

Independent claim 11 recites computer readable medium containing a computer program for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path. The computer program includes instructions for storing an access control statement in a cache entry (806a, 807a, 808a, 809a) for a path (806, 807, 808, 809) associated with a node of the plurality of nodes. *See, e.g.*, pg. 5, ln. 20 to pg. 6, ln. 2; pg. 18, lns. 14-18; figs. 1 and 8. The computer program also includes instructions for receiving a query, wherein the query comprises a request to access the node (702). *See, e.g.*, pg. 18, lns. 4-10; fig. 7A. In addition, the computer program includes instructions for checking the cache entry (806a, 807a, 808a, 809a) for the path (806, 807, 808, 809) associated with the node (704). *See, e.g.*, pg. 18, lns. 11-19; figs. 7A and 8. The computer program further includes instructions for granting or denying access to the node based on the access control statement in the cache entry (806a, 807a,

808a, 809a) for the path (806, 807, 808, 809) associated with the node (708, 718, 719, 730, 732). *See, e.g.*, pg. 18, ln. 20 to pg. 19, ln. 3; pg. 19, lns. 6-8; figs. 7A, 7B, and 8.

Independent claim 21 recites a method for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path. The method includes storing an access control statement in a cache entry (806a, 807a, 808a, 809a) for a path (806, 807, 808, 809) associated with a node of the plurality of nodes. *See, e.g.*, pg. 5, ln. 20 to pg. 6, ln. 2; pg. 18, lns. 14-18; figs. 1 and 8. The method also includes receiving a query, wherein the query comprises a request to access the node (702). *See, e.g.*, pg. 18, lns. 4-10; fig. 7A. In addition, the method includes checking the cache entry (806a, 807a, 808a, 809a) for the path (806, 807, 808, 809) associated with the node (704). *See, e.g.*, pg. 18, lns. 11-19; figs. 7A and 8. The method also includes granting access to the node responsive to the access control statement being a grant statement (718). *See, e.g.*, pg. 18, lns. 21-22; fig. 7A. Additionally, the method includes denying access to the node responsive to the access control statement being a deny statement (708). *See, e.g.*, pg. 18, lns. 20-21; fig. 7A. The method further includes evaluating a value expression for the path associated with the node to produce a result in response to the access control statement being an unknown statement or a data-dependent statement (716, 726), wherein the value expression is an executable statement based on an access control policy affecting the path and indicates who has access to the node. *See, e.g.*, pg. 10, lns. 7-10; pg. 18, ln. 22 to pg. 19, ln. 8; figs. 7A and 7B.

Independent claim 24 recites computer readable medium containing a computer program for performing path-level access control evaluation for a structured document, wherein the

structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path. The computer program includes instructions for storing an access control statement in a cache entry (806a, 807a, 808a, 809a) for a path (806, 807, 808, 809) associated with a node of the plurality of nodes. *See, e.g.*, pg. 5, ln. 20 to pg. 6, ln. 2; pg. 18, lns. 14-18; figs. 1 and 8. The computer program also includes instructions for receiving a query, wherein the query comprises a request to access the node (702). *See, e.g.*, pg. 18, lns. 4-10; fig. 7A. In addition, the computer program includes instructions for checking the cache entry (806a, 807a, 808a, 809a) for the path (806, 807, 808, 809) associated with the node (704). *See, e.g.*, pg. 18, lns. 11-19; figs. 7A and 8. The computer program also includes instructions for granting access to the node responsive to the access control statement being a grant statement (718). *See, e.g.*, pg. 18, lns. 21-22; fig. 7A. Additionally, the computer program includes instructions for denying access to the node responsive to the access control statement being a deny statement (708). *See, e.g.*, pg. 18, lns. 20-21; fig. 7A. The computer program further includes instructions for evaluating a value expression for the path associated with the node to produce a result in response to the access control statement being an unknown statement or a data-dependent statement (716, 726), wherein the value expression is an executable statement based on an access control policy affecting the path and indicates who has access to the node. *See, e.g.*, pg. 10, lns. 7-10; pg. 18, ln. 22 to pg. 19, ln. 8; figs. 7A and 7B.

Independent claim 27 recites a system for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path. The system includes a database management system (105) operable to receive a query, wherein the query comprises a request to access a node of the plurality of nodes (702). *See, e.g.*, pg. 18, lns. 4-10; figs. 1 and 7A. The

system also includes a cache (800) coupled to the database management system (105), the cache (800) being operable to store an access control statement in a cache entry (806a, 807a, 808a, 809a) for a path (806, 807, 808, 809) associated with the node, wherein the database management system (105) is further operable to check the cache entry (806a, 807a, 808a, 809a) for the path (806, 807, 808, 809) associated with the node and to grant or deny access to the node based on the access control statement in the cache entry (806a, 807a, 808a, 809a) for the path (806, 807, 808, 809) associated with the node (704, 708, 718, 719, 730, 732). *See, e.g.*, pg. 18, lns. 11-19; pg. 18, ln. 20 to pg. 19, ln. 3; pg. 19, lns. 6-8; figs. 1, 7A, 7B, and 8.

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

1. Appellant requests review as to claims 1-31, and their rejection under 35 U.S.C. § 102(b) as being anticipated by “Design and Implementation of an Access Control Processor for XML Documents” by Ernesto Damiani, et. al. (hereinafter “Damiani”).

VII. ARGUMENTS

1. Claims 1, 11, 21, 24, and 27 Are Not Anticipated by Damiani

Claim 1 recites a method for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path. The method includes storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes, receiving a query, wherein the query comprises a request to access the node, checking the cache entry for the path associated with the node, and granting or denying access to the node based on the access control statement in the cache entry for the path associated with the node.

Damiani does not disclose, teach, or suggest the claimed subject matter.

Damiani is directed to “an *Access Control System* for XML . . . [that] allow[s] for definition and enforcement of access restrictions directly on the structure and content of XML documents” (Abstract of Damiani).

(A)(i) Damiani does not disclose, teach, or suggest “storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes”

Damiani does not disclose, teach, or suggest “storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes,” as recited in claim 1.

In the final Office action, the Examiner states:

In response to Applicant’s argument that “Damiani does not disclose, teach, or suggest “storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes””, the arguments have been fully considered but are not deemed persuasive. Applicant argues that Damiani et al. teaches storing the “structured document” (XML file) and not the “access control statement”. The cache of Damiani et al. does not store the structured document in un-parsed, “plain-text” form (page 68, section 5.2, step 4, “Unparsing”); the structured document is “parsed, labeled, [and] transformed” and stored in a data structure when cached. See page 69, column 1, lines 2-5. The term “otherwise” implies that the cache stores documents in a data structure and not plain text. Fig. 5 clearly indicates the “transformed” document being parsed into trees comprised of nodes (see Figure 5, the tree to the left of box “3.transformation”).

(September 19, 2006 final Office action, pgs. 11-12).

Appellant is at a loss as to how storing a transformed DOM tree representation of an XML document in a cache in Damiani is in any way related to “storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes,” as recited in claim 1. In particular, claim 1 does not recite storing the nodes of a structured document in a cache, but rather, claim 1 recites “storing an access control statement . . . for a path associated

with a node.” As clearly shown in cache 800 in Figure 8 of the present application, it is the paths that are associated with the nodes of the structured document that are stored in cache 800 along with the control access statements, not the actual nodes of the structured document. Therefore, it is irrelevant that Damiani discusses storing a transformed DOM tree representation of an XML document in a cache.

In addition, it is unclear what element in Damiani the Examiner is construing as disclosing the “access control statement” recited in claim 1. If the Examiner is construing the “+” and “-” labels on the DOM tree in Figure 5 of Damiani as disclosing the “access control statement” recited in claim 1, Appellant respectfully reminds the Examiner that claim 1 recites “storing an access control statement . . . for a path associated with a node” (emphasis added). The discussion in Damiani regarding the “+” and “-” labels only relate to nodes of the DOM tree. Damiani does not disclose, teach, or suggest that the “+” and “-” labels are in any way related to a “path.”

Accordingly, Damiani does not disclose, teach, or suggest “storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes,” as recited in claim 1.

(A)(ii) Damiani does not disclose, teach, or suggest “checking the cache entry for the path associated with the node”

Damiani does not disclose, teach, or suggest “checking the cache entry for the path associated with the node,” as recited in claim 1.

In the final Office action, the Examiner states:

Damiani et al. teaches . . . checking the cache entry for the path associated with the node (see page 66, section 4, “Authorization enforcement”, lines 10-13 and page 68, section 5.3, “Performance and caching”, lines 11-12).

(September 19, 2006 final Office action, pg. 3).

The first passage of Damiani cited by the Examiner states:

The access control process must therefore evaluate the authorization applicable to an access request to compute such a view.

(Pg. 66, “Authorization enforcement”, Ins. 10-13). Although the cited passage of Damiani discusses evaluating an authorization applicable to an access request, nowhere does the cited passage of Damiani disclose, teach, or suggest that the evaluation involves “checking the cache entry for the path associated with the node,” as recited in claim 1. In fact, the terms “cache” and “path associated with the node” are nowhere to be found in the cited passage.

The second passage of Damiani cited by the Examiner states:

When the request comes, the cache is searched. If an instance of the requested document for the same subject is found in the cache, then the cache copy is served. Otherwise, the document is parsed, labeled, transformed, unparsed and sent to the client; also, the transformed document is stored into the cache.

(Pgs. 68-69, “Performance and caching”, Ins. 11-17). As discussed above, Damiani only discusses caching a transformed DOM tree representation of an XML document. Damiani does not disclose, teach, or suggest storing a path associated with a node of a structured document in a

cache. Hence, it necessarily follows that Damiani does not disclose, teach, or suggest “checking the cache entry for the path associated with the node,” as recited in claim 1, because there aren’t any paths in the cache in Damiani.

Therefore, Damiani also fails to disclose, teach, or suggest “checking the cache entry for the path associated with the node,” as recited in claim 1.

(A)(iii) Damiani does not disclose, teach, or suggest “granting or denying access to the node based on the access control statement in the cache entry for the path associated with the node”

Damiani does not disclose, teach, or suggest “granting or denying access to the node based on the access control statement in the cache entry for the path associated with the node,” as recited in claim 1.

In the final Office action, the Examiner states:

In response to Applicant’s arguments that Damiani et al. “does not disclose, teach, or suggest “granting or denying access to the node based on the access control statement in the cache entry for the path associated with the node,” as recited in claim 1, since it only discusses using DTD-level and document-level authorizations to determine what a requester sees”, the arguments have been fully considered but are not deemed persuasive. After the cited portion of the reference, Damiani et al. goes on to say “the analysis [...] produces an access decision (access or not access) **on each node** of the document” (emph. added. See page 66, section 4, lines 16-19). Section 4, when considered in its entirety, contains several such references to per-node access control assignments. For example, see page 66, column 2, number (1), “Authorizations on **a** node”, emph. added.

(September 19, 2006 final Office action, pg. 12).

Regardless of whether Damiani mentions “per-node access control assignments,” claim 1 recites “granting or denying access to the node based on the access control statement in the cache entry for the path associated with the node” (emphasis added). As discussed above, Damiani

does not disclose, teach, or suggest “storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes,” as recited in claim 1. Thus, it necessarily follows that Damiani fails to disclose, teach, or suggest “granting or denying access to the node based on the access control statement in the cache entry for the path associated with the node,” as recited in claim 1 (emphasis added), because there aren’t any “access control statement[s]” nor any “path[s] associated with . . . node[s]” of a structured document stored in the cache of Damiani.

Consequently, it is irrelevant whether Damiani mentions “per-node access control assignments” because even assuming argumentatively that Damiani discloses granting or denying of access on a node-by-node basis, the grant or denial of access will not be “based on the access control statement in the cache entry for the path associated with the node,” as recited in claim 1.

Accordingly, Damiani does not disclose, teach, or suggest “granting or denying access to the node based on the access control statement in the cache entry for the path associated with the node,” as recited in claim 1.

(A)(iv) The Examiner has not established anticipation under 35 U.S.C. § 102

Anticipation under 35 U.S.C. § 102 requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention. (*See, e.g., Electro Med. Sys. S.A. v. Cooper Life Sciences*, 34 F.3d 1048, 32 U.S.P.Q.2d 1017, 1019 (Fed. Cir. 1994)). The Examiner has failed to show that the elements discussed in sections (A)(i) and (A)(iii) above are disclosed in Damiani.

Therefore, claim 1, and the claims that depend therefrom, are not anticipated by Damiani. Given that claims 11, 21, 24, and 27 each recite elements similar to those of claim 1, claims 11, 21, 24, and 27, and the claims that depend therefrom, are not anticipated by Damiani for at least the same reasons.

2. Claims 21 and 24 Are Further Not Anticipated by Damiani

Claim 21 recites a method for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path. The method includes storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes, receiving a query, wherein the query comprises a request to access the node, checking the cache entry for the path associated with the node, granting access to the node responsive to the access control statement being a grant statement, denying access to the node responsive to the access control statement being a deny statement, and evaluating a value expression for the path associated with the node to produce a result in response to the access control statement being an unknown statement or a data-dependent statement, wherein the value expression is an executable statement based on an access control policy affecting the path and indicates who has access to the node.

Damiani does not disclose, teach, or suggest the claimed subject matter.

Damiani is directed to “an *Access Control System* for XML . . . [that] allow[s] for definition and enforcement of access restrictions directly on the structure and content of XML documents” (Abstract of Damiani).

(B)(i) Damiani further fails to disclose, teach, or suggest “evaluating a value expression for the path associated with the node to produce a result in

response to the access control statement being an unknown statement or a data-dependent statement, wherein the value expression is an executable statement based on an access control policy affecting the path and indicates who has access to the node”

As discussed above in sections (A)(i)-(A)(ii), Damiani does not disclose, teach, or suggest “storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes” or “checking the cache entry for the path associated with the node,” as recited in claim 1, which are similarly recited in claim 21.

Damiani further fails to disclose, teach, or suggest “evaluating a value expression for the path associated with the node to produce a result in response to the access control statement being an unknown statement or a data-dependent statement, wherein the value expression is an executable statement based on an access control policy affecting the path and indicates who has access to the node,” as recited in claim 21.

In the final Office action, the Examiner states:

Damiani et al. teaches . . . evaluating a value expression for the path associated with the node to produce a result in response to the access control statement being an unknown statement or a data-dependent statement (see Examiner’s comments regarding claim 2),

wherein the value expression is an executable statement based on an access control policy affecting the path and indicates who has access to the node (see Examiner’s comments regarding claim 1).

(September 19, 2006 final Office action, pgs. 7-8).

With regards to claim 2, the Examiner states:

Damiani et al. teaches wherein the access control statement is one of a grant statement (see page 66, section 4, “Authorization enforcement” line 32 “+’ (permission)”), a deny statement (“-’ (denial)”), an unknown statement (line 33, “ε’ (no authorization)”) and a data-dependent statement (see page 63, section 3, “Authorizations”, bullet-point 1, where “data-dependent statement” is read on “specific documents”).

(September 19, 2006 final Office action, pg. 4).

Claims 1 and 2, however, do not recite “a value expression,” as recited in claim 21. It appears that the Examiner is relying upon the same passages in Damiani as disclosing both “an access control statement” and “a value expression.” As clearly recited in claim 21, described in the specification, and shown in Figure 6 and 8 of the drawings, “an access control statement” is not the same as “a value expression.” Therefore, the Examiner cannot rely upon the “+” and “-” labels in Damiani as disclosing both the “access control statement” and the “value expression.”

Further, claim 21 specifically defines a “value expression” as “an executable statement.” Appellant respectfully submits that one of ordinary skill in the art would not construe the “+” and “-” labels in Damiani as executable statements.

Accordingly, Damiani does not disclose, teach, or suggest “evaluating a value expression for the path associated with the node to produce a result in response to the access control statement being an unknown statement or a data-dependent statement, wherein the value expression is an executable statement based on an access control policy affecting the path and indicates who has access to the node,” as recited in claim 21.

(B)(ii) The Examiner has not established anticipation under 35 U.S.C. § 102

Anticipation under 35 U.S.C. § 102 requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention. (*See, e.g., Electro Med. Sys. S.A. v. Cooper Life Sciences*, 34 F.3d 1048, 32 U.S.P.Q.2d 1017, 1019 (Fed. Cir. 1994)). The Examiner has failed to show that the element discussed in section (B)(i) above is disclosed in Damiani.

Therefore, claim 21, and the claims that depend therefrom, are further not anticipated by Damiani. Given that claim 24 recites elements similar to those of claim 21, claim 24, and the claims that depend therefrom, are further not anticipated by Damiani for at least the same reasons.

CONCLUSION

On the basis of the above remarks, Appellant respectfully submits that the final rejection should be reversed.

Respectfully submitted,
SAWYER LAW GROUP LLP



Dated: April 16, 2006

Erin C. Ming
Attorney for Appellant
Reg. No. 47,797
(650) 475-1449

APPENDIX OF CLAIMS

1. (Previously Presented) A method for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path, the method comprising the steps of:

- (a) storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes;
- (b) receiving a query, wherein the query comprises a request to access the node;
- (c) checking the cache entry for the path associated with the node; and
- (d) granting or denying access to the node based on the access control statement in the cache entry for the path associated with the node.

2. (Previously Presented) The method of claim 1, wherein the access control statement is one of a grant statement, a deny statement, an unknown statement, and a data dependent statement.

3. (Previously Presented) The method of claim 2, wherein step (d) further comprises:

- (d1) granting access to the node responsive to the access control statement being a grant statement.

4. (Previously Presented) The method of claim 2, wherein step (d) further comprises:

- (d1) denying access to the node responsive to the access control statement being a deny statement.

5. (Previously Presented) The method of claim 2, wherein step (d) further comprises:

(d1) evaluating an access control policy affecting the path in response to the access control statement being an unknown statement;

(d2) granting access responsive to a result of the evaluation granting access; and

(d3) denying access responsive to the result of the evaluation denying access.

6. (Previously Presented) The method of claim 5, further comprising:

(e) determining whether the access control policy affecting the path is data dependent;

(f) changing the access control statement in the cache entry from the unknown statement to a grant statement or a deny statement based on the evaluation in response to the access control policy being data independent; and

(g) changing the access control statement in the cache entry from the unknown statement to a data-dependent statement in response to the access control policy being data dependent.

7. (Previously Presented) The method of claim 2, wherein step (d) further comprises:

(d1) evaluating an access control policy affecting the path in response to the access control statement being a data-dependent statement;

(d2) granting access responsive to a result of the evaluation granting access; and

(d3) denying access responsive to the result of the evaluation denying access.

8. (Previously Presented) The method of claim 1, further comprising:
 - (e) repeating steps (c) and (d) for a next node in the plurality of nodes.
9. (Original) The method of claim 5, wherein evaluating step (d1) further comprises:
 - (d1i) evaluating a value expression for the path associated with the node, wherein the value expression is an executable statement based on the access control policy affecting the path and indicates who has access to the node.
10. (Previously Presented) The method of claim 1, wherein steps (c) and (d) are performed during run-time.
11. (Previously Presented) A computer readable medium containing a computer program for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path, the computer program comprising programming instructions for:
 - (a) storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes;
 - (b) receiving a query, wherein the query comprises a request to access the node;
 - (c) checking the cache entry for the path associated with the node; and
 - (d) granting or denying access to the node based on the access control statement in the cache entry for the path associated with the node.

12. (Previously Presented) The computer readable medium of claim 11, wherein the access control statement is one of a grant statement, a deny statement, an unknown statement, and a data dependent statement.

13. (Previously Presented) The computer readable medium of claim 12, wherein instruction (d) further comprises:

(d1) granting access to the node responsive to the access control statement being a grant statement.

14. (Previously Presented) The computer readable medium of claim 12, wherein instruction (d) further comprises:

(d1) denying access to the node responsive to the access control statement being a deny statement.

15. (Previously Presented) The computer readable medium of claim 12, wherein instruction (d) further comprises:

(d1) evaluating an access control policy affecting the path in response to the access control statement being an unknown statement;

(d2) granting access responsive to a result of the evaluation granting access; and

(d3) denying access responsive to the result of the evaluation denying access.

16. (Previously Presented) The computer readable medium of claim 15, wherein the computer program further comprises programming instructions for:

- (e) determining whether the access control policy affecting the path is data dependent;

- (f) changing the access control statement in the cache entry from the unknown statement to a grant statement or a deny statement based on the evaluation in response to the access control policy being data independent; and

- (g) changing the access control statement in the cache entry from the unknown statement to a data-dependent statement in response to the access control policy being data dependent.

17. (Previously Presented) The computer readable medium of claim 12, wherein instruction (d) further comprises:

- (d1) evaluating an access control policy affecting the path in response to the access control statement being a data-dependent statement;

- (d2) granting access responsive to a result of the evaluation granting access; and

- (d3) denying access responsive to the result of the evaluation denying access.

18. (Previously Presented) The computer readable medium of claim 11, wherein the computer program further comprises programming instructions for:

- (e) repeating instructions (c) and (d) for a next node in the plurality of nodes.

19. (Original) The computer readable medium of claim 15, wherein evaluating instruction (d1) further comprises:

(d1i) evaluating a value expression for the path associated with the node, wherein the value expression is an executable statement based on the access control policy affecting the path and indicates who has access to the node.

20. (Previously Presented) The computer readable medium of claim 11, wherein instructions (c) and (d) are performed during run-time.

21. (Previously Presented) A method for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path, the method comprising the steps of:

(a) storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes, wherein the access control statement is one of a grant statement, a deny statement, an unknown statement, and a data-dependent statement;

(b) receiving a query, wherein the query comprises a request to access the node;

(c) checking the cache entry for the path associated with the node;

(d) granting access to the node responsive to the access control statement being a grant statement;

(e) denying access to the node responsive to the access control statement being a deny statement; and

(f) evaluating a value expression for the path associated with the node to produce a result in response to the access control statement being an unknown statement or a data-dependent statement,

wherein the value expression is an executable statement based on an access control policy affecting the path and indicates who has access to the node.

22. (Previously Presented) The method of claim 21, further comprising:

(g) granting or denying access to the node based on the result of the evaluation;

(h) changing the access control statement in the cache entry from the unknown statement to a grant statement or a deny statement based on the result of the evaluation in response to the access control policy being data independent; and

(i) changing the access control statement in the cache entry from the unknown statement to a data-dependent statement in responsive to the access control policy being data dependent.

23. (Previously Presented) The method of claim 22, further comprising:

(j) repeating steps (c) through (i) for a next node in the plurality of nodes.

24. (Previously Presented) A computer readable medium containing a computer program for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path, the computer program comprising programming instructions for:

(a) storing an access control statement in a cache entry for a path associated with a node of the plurality of nodes, wherein the access control statement is one of a grant statement, a deny statement, an unknown statement, and a data-dependent statement;

(b) receiving a query, wherein the query comprises a request to access the node;

(c) checking the cache entry for the path associated with the node;

(d) granting access to the node responsive to the access control statement being a grant statement;

(e) denying access to the node responsive to the access control statement being a deny statement; and

(f) evaluating a value expression for the path associated with the node to produce a result in response to the access control statement being an unknown statement or a data-dependent statement,

wherein the value expression is an executable statement based on an access control policy affecting the path and indicates who has access to the node.

25. (Previously Presented) The computer readable medium of claim 24, wherein the computer program further comprises programming instructions for:

(g) granting or denying access to the node based on the result of the evaluation;

(h) changing the access control statement in the cache entry from the unknown statement to a grant statement or a deny statement based on the result of the evaluation in response to the access control policy being data-independent; and

(i) changing the access control statement in the cache entry from the unknown statement to a data-dependent statement in response to the access control policy being data dependent.

26. (Previously Presented) The computer readable medium of claim 25, wherein the computer program further comprises programming instructions for:

(j) repeating instructions (c) through (i) for a next node in the plurality of nodes.

27. (Previously Presented) A system for performing path-level access control evaluation for a structured document, wherein the structured document comprises a plurality of nodes and each of the plurality of nodes is described by a path, the system comprising:

a database management system operable to receive a query, wherein the query comprises a request to access a node of the plurality of nodes; and

a cache coupled to the database management system, the cache being operable to store an access control statement in a cache entry for a path associated with the node,

wherein the database management system is further operable to check the cache entry for the path associated with the node and to grant or deny access to the node based on the access control statement in the cache entry for the path associated with the node.

28. (Previously Presented) The system of claim 27, wherein the access control statement is one of a grant statement, a deny statement, an unknown statement, and a data dependent statement.

29. (Previously Presented) The system of claim 28, further comprising:

an access control mechanism coupled to the database management system, the access control mechanism being operable to determine access control to the node responsive to the access control statement being an unknown statement or a data-dependent statement.

30. (Previously Presented) The system of claim 29, wherein the access control mechanism is further operable to generate a value expression for the path associated with the node based on an access control policy affecting the path, and wherein the database management system is further operable to evaluate the value expression for the path to determine whether to grant or deny access to the node.

31. (Previously Presented) The system of claim 30, wherein the database management system is further operable to change the access control statement in the cache entry from the unknown statement to a grant statement or a deny statement based on a result of the evaluation of the value expression responsive to the value expression for the path being data-independent and to change the access control statement in the cache entry from the unknown statement to a data dependent statement responsive to the value expression for the path being data-dependent.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None